Interfacing a Cognitive Agent Platform with a Virtual World: a Case Study using Second Life

(Extended Abstract)

Surangika Ranathunga   Stephen Cranefield   Martin Purvis

Department of Information Science
University of Otago
Dunedin 9054, New Zealand
{surangika, scranefield, mpurvis}@infoscience.otago.ac.nz

ABSTRACT

Online virtual worlds provide a rich platform for remote human interaction, and are increasingly being used as a simulation platform for multi-agent systems and as a way for software agents to interact with humans. It would therefore be beneficial to provide techniques allowing high-level agent development tools, especially cognitive agent platforms such as belief-desire-intention (BDI) programming frameworks, to be interfaced with virtual worlds. This is not a trivial task as it involves mapping potentially unreliable sensor readings from complex virtual environments to a domain-specific abstract logical model of observed properties and/or events. This paper investigates this problem in the context of agent interactions in a multi-agent system simulated in Second Life. We present a framework which facilitates the connection of any multi-agent platform with Second Life, and demonstrate it in conjunction with the Jason BDI interpreter.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents, Multiagent systems

General Terms

Design, Experimentation

Keywords

Multiagent systems, BDI agents, Jason, Second Life

1. INTRODUCTION

Multi-purpose online virtual worlds provide a sophisticated and convenient simulation platform for testing multi-agent systems and other AI concepts, where software-controlled agents can be made to interact with human-controlled agents. It would therefore be beneficial to provide techniques allowing high-level agent development tools, especially cognitive agent platforms such as belief-desire-intention (BDI) programming frameworks, to be interfaced with virtual worlds.

When interfacing agent platforms with virtual worlds, there are two non-trivial challenges to be addressed: how the agent actions are performed on the virtual environment and how the large volumes of (potentially unreliable) sensor readings from the virtual environment are mapped to a domain-specific abstract logical model of observed properties and/or events, to be used by a multi-agent system.

This paper addresses these challenges in the context of agent interactions in a multi-agent system simulated in the popular multi-purpose virtual world Second Life[1]. The main focus of this paper is on how the potentially unreliable data received by an agent deployed in a Second Life simulation can be processed to create a domain-specific high-level abstract model to be used by the agent's cognitive modules. In order to accomplish this, we have developed a framework with the use of the *LIBOMV* client library[2], and this framework facilitates the connection of any multi-agent framework with Second Life. The main responsibility of the framework is to accurately extract the sensor readings from Second Life, to identify the high-level domain specific information embedded in those low-level data, and finally to convert this information into a form that can be used by the multi-agent system. Here, the latter two aspects have not gained much attention in research related to Second Life.

2. SYSTEM DESIGN

Figure 1 shows how the different components of the system are interfaced with each other. In this paper, we demonstrate our framework in conjunction with the Jason BDI agent development platform [2].

2.1 Interface Between the LIBOMV Client and the Jason Agent

The interface between the LIBOMV client and the Jason agent is facilitated using sockets (denoted by 'S' in Figure 1). This decoupling makes it possible to connect any agent platform with the LIBOMV clients. The module that contains LIBOMV clients is capable of handling multiple concurrent LIBOMV clients and socket connections. Therefore if the corresponding multi-agent system is able to create concurrently operating agents, this can easily create a multi-agent simulation inside Second Life. Consequently, the module that contains the Jason platform is designed in such a way that it is capable of handling multiple concurrent instances of socket

Cite as: Interfacing a Cognitive Agent Platform with a Virtual World: a Case Study using Second Life (Extended Abstract), S. Ranathunga, S. Cranefield, M. Purvis, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1181-1182.

[1] `http://secondlife.com`

[2] `http://lib.openmetaverse.org/wiki/Main_Page`

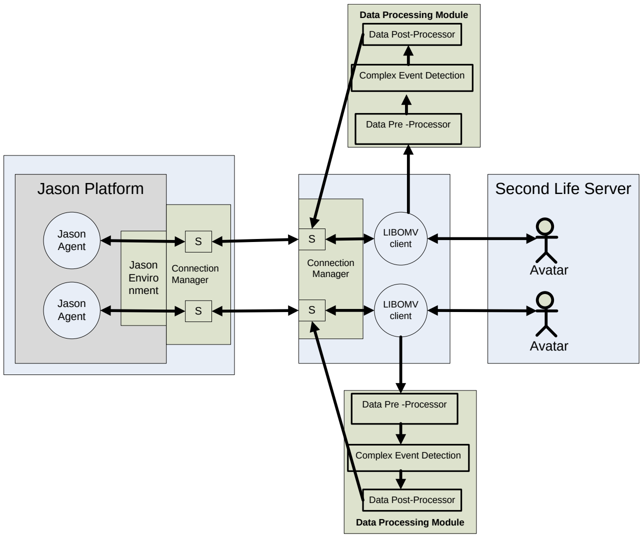

Figure 1: Overall System Design

connections connected to the Jason agents.

2.2 Interface Between the LIBOMV Client and the Second Life Server

Although a LIBOMV client connected to Second Life can extract data from Second Life in a more robust way than using the *Linden Scripting Language (LSL)* , it also has several limitations, which affect the accuracy of the extracted sensory readings. Therefore we implemented a combined approach to extract data from Second Life, where a scripted object is attached to the LIBOMV client. Detection of the avatars and objects to be monitored is done at the LIBOMV client side. Identification information for these is then sent to the script. As the script already knows what to be tracked, an efficient, light-weight function can be used to record the position and velocity information instead of the normal LSL sensor functionality. Avatar animation updates are directly captured by the LIBOMV client to make sure animations with short durations (eg. crying or blowing a kiss) are not missed out. The communication messages (chat exchanged in the public chat channels, instant messages sent to the agent) are also directly captured by the LIBOMV client.

2.3 Data Processing Module

The data processing module consists of three main components; the data pre-processor, the complex event detection module and the data post-processor. The responsibility of the data processing module is to map the received sensor readings from complex Second Life environments to a domain-specific abstract logical model. In essence, it creates snapshots of the system that include low-level data (position and animation information of the avatars and objects) generated in the given Second Life environment in a given unit of time, along with the identified high-level domain-specific information and other contextual information.

In accomplishing this, first the data pre-processor amalgamates the data received from the LSL script and the received updates for avatar animations and communication messages, and creates snapshots of the environment. A snapshot includes the position and velocity information of all the avatars and objects of interest that are valid at a given instant of time, along with avatar animation information. The data pre-processor also deduces the basic high-level information about the avatars and objects, e.g. whether an avatar is moving, and if so, in which direction and the movement type (e.g. walking, running or flying), and whether an avatar is in close proximity to another avatar or an object of interest. Other contextual information such as the location of the avatar or the role it is playing can also be attached to this retrieved information as needed.

These low level data are then sent to the complex event detection module, to identify the high-level domain-specific information embedded in those low-level data. For this, we use an event stream processing engine called Esper[3].

Finally, the data post-processor converts the processed data into an abstract model to be passed to the connected multi-agent system. The detected low-level and high-level events, along with other context information are grouped into states (a state corresponds a snapshot of the Second Life environment at a given instant of time) which are represented as a set of propositions. These propositions are sent to the multi-agent system, to be converted to any representation needed by the multi-agent system. For example, in Jason, these are converted to percepts, which are recorded as agent beliefs.

3. CONCLUSION

In this paper we presented a framework that can be used to deploy multiple concurrent agents in complex Second Life simulations, and demonstrated it with the Jason BDI agent development platform. The main focus of this paper was on how the potentially unreliable data received by an agent deployed in a Second Life simulation should be processed to create a domain-specific high-level abstract model to be used by the agent's cognitive modules. Although there have been some practical implementations of agent societies inside Second Life [1], they have mainly focused on creating Second life simulations specifically for human-agent interaction, rather than trying to integrate agent platforms with the already existing Second Life simulations as we have done. Moreover, we do not see these specific problems have been properly investigated there. On the other hand, the other theoretical proposals that addressed this issue have not been implemented yet [3].

We have successfully tested our framework with Jason agents deployed in the SecondFootball[4] simulation in Second Life, and currently the framework is customized for this simulation. However in the future, we are planning to make the framework more generalized. Further details, discussion and a comparison with related work can be found in the full version of this paper [4].

4. REFERENCES

[1] A. Bogdanovych, S. Simoff, and M. Esteva. Virtual institutions: Normative environments facilitating imitation learning in virtual agents. In *Intelligent Virtual Agents*, volume 5208 of *Lecture Notes in Computer Science*, pages 456–464. Springer, Berlin, Heidelberg, 2008.

[2] R. H. Bordini, J. F. Hubner, and M. Wooldridge. *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley & Sons Ltd, England, 2007.

[3] D. J. H. Burden. Deploying embodied AI into virtual worlds. *Knowledge-Based Systems*, 22(7):540–544, 2009.

[4] S. Ranathunga, S. Cranefield, and M. Purvis. Interfacing a cognitive agent platform with Second Life. Discussion Paper 2011/03, Department of Information Science, University of Otago, 2011. `http://eprints.otago.ac.nz/1093/`.

[3] `http://esper.codehaus.org`

[4] `http://www.secondfootball.com`